PARALLEL PROCESSING OF FINITE STRAIN, MATERIALLY NONLINEAR AND INCOMPRESSIBLE FINITE ELEMENT ANALYSIS PROBLEMS

Ahmet T. Becene[a] and **Renato Perucchio**[b]

[a]Delphi
5500 W. Henrietta Road, MC 146.HEN.540
Rochester, NY 14602
ahmet.becene@delphi.com

[b]Department of Mechanical Engineering
The University of Rochester
Rochester, NY
rlp@me.rochester.edu

This paper addresses the problem of parallel processing of finite strain, materially nonlinear, and incompressible finite element analysis problems on shared memory and distributed memory computing platforms.

We employ shared memory parallel version of the SuperLU [1] libraries as the shared memory parallelization tool. Initially, extending an earlier implementation of *Updated Lagrangian* (UL) FE formulation developed at the University of Rochester [2], we implemented the SuperLU libraries during the solution of the linearized equations step. The remaining steps of the FE solution procedure, including the elemental stiffness computations, follow the existing serial computing algorithm. Although this approach resulted in considerable speedups during the solution of the linearized equations step, the overall solution times improved only marginally.

We then introduced distributed memory parallelization, requiring partitioning of the FE mesh. We implemented an automatic domain partitioning algorithm using the Metis [3] libraries. The partitioned domains provide the local data required for the domain decomposition parallel version of our nonlinear finite element algorithm. Numerical tests with unstructured hexahedral meshes show that the PartMeshNodal routine of Metis results in balanced partitions with roughly the same number of nodes in each domain.

For our distributed memory parallel algorithm, we implemented the domain decomposition method with a direct solver. MPICH [4] was used as the communications standard. Data structures from the serial version of the algorithm were expanded to include the required parameters for the parallel version. Elemental stiffness computations and stress recovery steps were executed in parallel. The resultant interface system was solved on a single processor. Since most of the steps in this algorithm run in parallel this approach resulted in higher speedups than those produced by our shared memory parallel algorithm.

References

[1] J. W. Demmel, J. Gilbert and X. S. Li, “SuperLU Users’ Guide" *Techreport*, CSD-97-944, 1997.
[2] Srinivasan, R. and Perucchio, R., "Finite element analysis of anisotropic nonlinear incompressible elastic solids by a mixed model" Int. J. Numer. Methods Eng., v 37, 3075–3092, 1994.
[3] G. Karypis and V. Kumar, “Metis, A Software for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0" 1998.
[4] W. D. Gropp and E. Lusk, "User’s Guide for a Portable Implementation of MPI" 1996